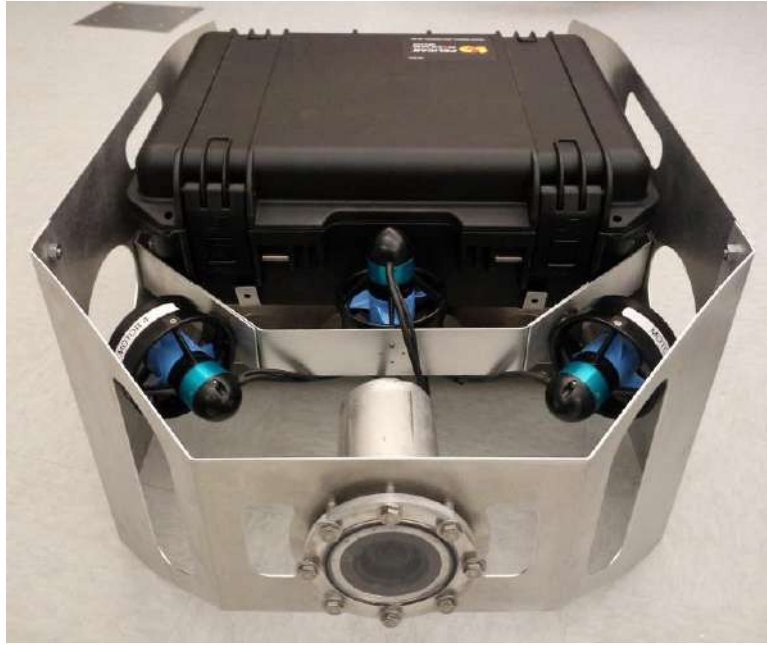


YellowFin

Embry-Riddle Aeronautical University Autonomous Underwater Vehicle



<http://roboticsassociation.org/robosub.html>

Team Leader:

Evan Williams

Team Members:

Stephen Cronin, Robert Goring

Faculty Advisors:

Dr. Charles Reinholtz

Professor and Chair - Mechanical Engineering Department

Dr. Brian Butka

Professor - Electrical, Computer, Software, and Systems Engineering Department

&

Dr. Timothy A. Wilson

Professor and Chair - Electrical, Computer, Software, and Systems Engineering Department

**Embry Riddle Aeronautical University
600 S. Clyde Morris Blvd.
Daytona Beach, FL 32114**

Abstract

The Robotics Association at Embry-Riddle (RAER) is proud to present its entry, YellowFin, for the 19th Annual International RoboSub Competition. The year's RoboSub team consists of several new members with mentorship being provided by students and faculty members who are well versed in the unmanned and autonomous systems regime. The team working on YellowFin consists of a handful of undergraduate and graduate students representing different majors from the College of Engineering at Embry-Riddle. The RoboSub competition consists of several tasks, which require navigation, image processing, and object manipulation. These tasks model the applications of current commercially available AUVs. The Embry-Riddle RoboSub team has designed a new vehicle from the ground up for this year's competition.

1. Introduction

The Association of Unmanned Vehicle Systems International (AUVSI) and Office of Naval Research (ONR) are hosting the seventeenth annual international RoboSub Competition to be held in San Diego, California at the SPAWAR Transdec facility. The goal of this competition is to create a vehicle that can autonomously complete a set of tasks, which are mimic the abilities of current autonomous vehicles in industry. To create a vehicle and system capable of accomplishing all of these tasks, requires years of development and testing. Recognizing the concept of complexity through iteration, the RAER RoboSub team has devised the goal of creating a simple system that can perform the required tasks, with room for future modifications, as determined from competition performance.

For this year's entry the YellowFin system has been mechanically overhauled for a more efficient design with the capacity to support growth in hardware capacity. The new frame is designed to be light and portable, while maintaining structural integrity and performance. This engineering decision has provided the team with a robust, mobile and durable platform.

2. Hardware and Mechanics

The goal of last year's sub was to create a backpack portable submarine. This platform achieved its goal, however, we rapidly reached limitations when attempting to add hardware to accomplish tasks previously not attempted. A new vehicle was designed to address this issue and give us room to scale vehicle capabilities in the future. The old and new vehicle frames can be seen in Figure 1.

The electronics, computer, and the majority of our sensors are housed in a pelican case mounted at the center of the vehicle. This packaging allows for sensitive components to be transported separately from the frame.

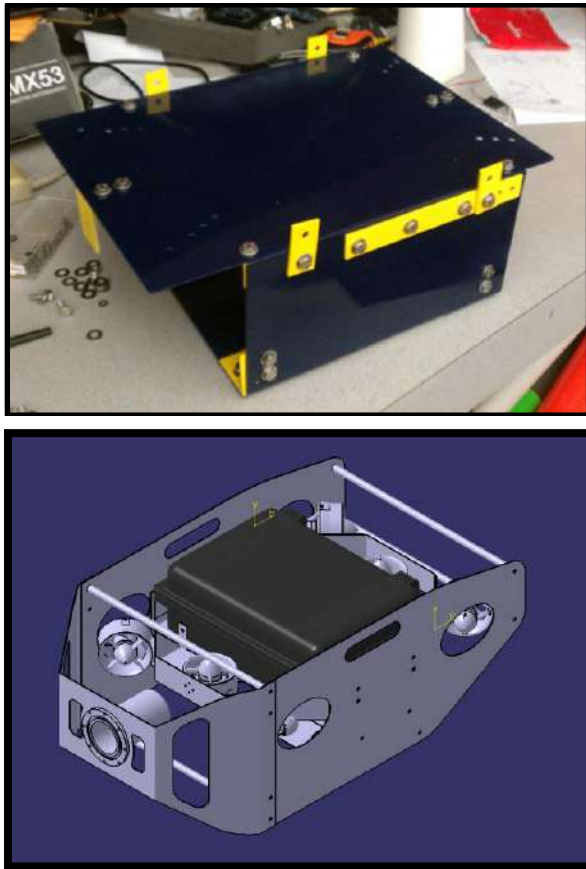


Figure 1: Old frame (top) and new frame render (bottom)

2.1 Frame / Structure

A significant change in this year's platform is the overhaul of the frame. The platform is designed to allow for growth in the platform

and introduce a holonomic thruster configuration, allowing for more degrees of freedom in motion as compared to the last vehicle.

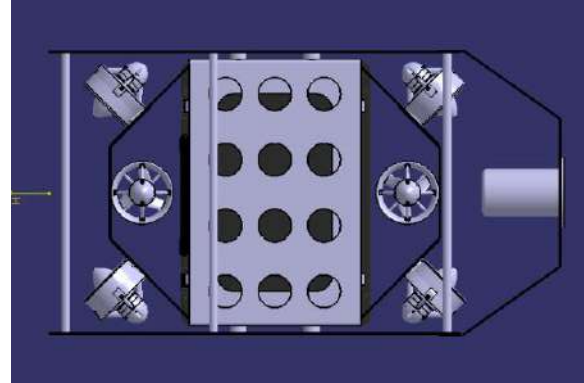


Figure 2: Cad rendering of frame (bottom)

Mechanically, the frame is constructed out of 16th inch thick aluminum. The frame is comprised of 5 main components: two side walls designed for easier transportation of the vehicle and protecting exterior components during transportation; the front panel, designed for mounting various devices including cameras and torpedo launchers; the mount on which the Pelican Storm m2300 case can securely attach to and finally the weight tray which mounts below the Pelican case for purposes of trimming the vehicle and achieving neutral buoyancy.

The vehicle is designed to be statically stable, which simplifies maneuvers in the water. Majority of the vehicle's weight is concentrated on the weight tray, giving YellowFin a low center of gravity, thus reducing pitch and roll. The weight tray also allows for weights to be adjusted, allowing for the vehicle to be statically balanced with different internal configurations and external mounts.

2.2 Internal Layout

All components for the system with the exception of the thrusters, hydrophones and camera are housed within the main hull.



Figure 3 - Vehicle Internal Layout

The ethernet cameras are housed in separate metal housings. The hydrophones are mounted in a separate bracket for purposes of positioning them for better signal clarity.

2.3 Thrusters

This iteration of YellowFin uses six BlueRobotics T200 thrusters: two in-line vertical and four horizontal, in a holonomic configuration with thrusters mounted at 45 degree angles. This configuration provides five degrees of freedom: translation in all planes, yaw, and pitch. This setup was determined to have the optimal level of maneuvering freedom to perform most tasks as the ability to roll was deemed unnecessary.

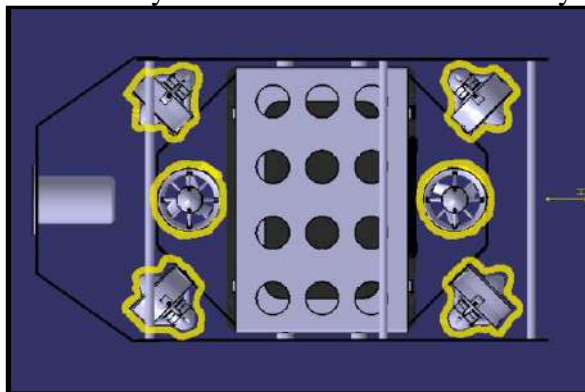


Figure 4 - Thruster Positions

Current motor configuration allows for simplified control algorithms. In this setup, motors are placed symmetrical to each other to prevent unexpected movements. To achieve reliable motion in the horizontal

plane, only the side thrusters need to be balanced.

2.4 Connectors

For connectors to the motors and cameras IP68 rated in-line and bulkhead connectors from LEMO are used. Each thruster uses six conductors: power and ground, and SDA, SCL, PWM, and logical ground - the motors each have a dedicated connector. The cameras use 8-pin LEMO connectors. All of the connectors used are rated for 150 meters. The primary tether is a wet-pluggable SubConn bulkhead and inline connector, rated for deep sea use.



Figure 5 - External LEMO Connectors

2.5 Camera Housing

Two matching custom housings are used to encase the camera systems. These housings were manufactured by 1/4" thick aluminum tubing, as well as 1/4" acrylic. A custom 3D printed mount has been created to hold the camera in place. The cameras are mounted so that one faces forward, while the other faces straight down. This allows for the system to visually inspect objects in front of the vehicle as well as those below.



Figure 6 - Camera Housing

2.6 Current Development

YellowFin does not yet have systems for completing all of the tasks. Currently, the droppers and torpedo launchers are being developed.

The dropper system in development consists of small 3D-printed torpedoes that are released by a servo. These 3D-printed torpedoes are hollow so that they can be weighted to fall straight.

The torpedo system consists of Toypedos that are spring launched and released by a servo. Having them spring launched ensures that they are released at a safe speed and won't cause injury to the pool or diver.

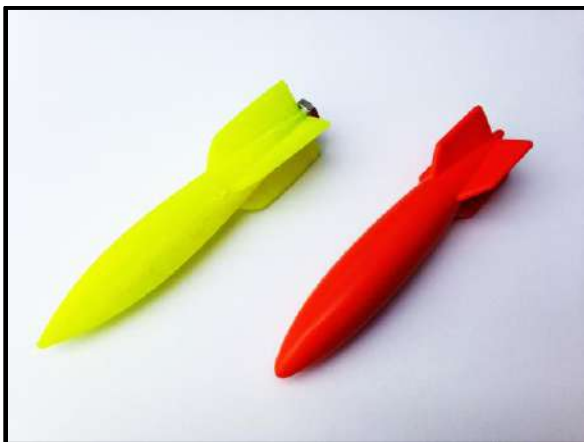


Figure 7 - Torpedoes for Dropping (left) and Firing (right)

3. Electronics

YellowFin’s electronic system consists of purchased, donated, and homemade components. The electronics are designed to be compact and modular. All of the components that are not a part of the computer are mounted to the top of the computer case seen in Figure 8 using Velcro.

3.1 Computer



Figure 8 – Computer case with Cover Removed

The computer consists of entirely off-the-shelf components. The computer specifications are seen below in Table 1.

Table 1 - Computer Specifications

Motherboard	Asus H81T/CSM
Processor	Intel i7-4790S @3.2 GHz
Memory	16GB DDR3 SoDimm
Storage	256 GB Solid State
Power Supply	Integrated 19V DC Input

3.2 Computer Voltage Converter

The power supply is given by two separate systems: a 6 cell Lithium-Polymer battery which outputs 25 volts, and a 5 cell Lithium-Polymer battery which outputs 21V. The first of these batteries supplies power to various subsystems of differing voltages, for which the voltage converters are listed below.

The latter of the two batteries is used exclusively to power the thrusters, which

carries the benefit of isolating the innately noisy motors and their controller board.

Table 2 – Voltage Converters

24VDC	19V DC (Motherboard input)
24VDC	48V DC (PoE capable Switch)



Figure 9 - Voltage Converter

3.3 VectorNav Compass

Next Year uses a VN-100 Rugged Inertial Measurement Unit (IMU) to navigate in consistent directions. This IMU was generously donated to RAER by VectorNav.



Figure 10 - VectorNav VN-100

The data used from this sensor are the Euler Angles of the sub, yaw, pitch, and roll. The main features of this IMU are its .02° RMS precision measurements, simple 2D magnetic field calibration, and ability to automatically filter out soft and hard iron disturbances. These filtering capabilities are vital when operating in close proximity to large switching currents, such as the motor controller.

3.4 Motor Controller



Figure 11: Intel Galileo

The motor controller is an off the shelf Intel Galileo board. In order to communicate with the motors, the I²C protocol is utilized. Each motor is individually addressed and a single connection for all six thrusters is brought into the board. The Galileo communicates with the computer through UDP. The motor controller itself is powered through the power over ethernet, a module was added to the Galileo to make this possible.

3.6 Hydrophones

Four Teledyne Reson 4013 hydrophones are grouped into two pairs. These pairs are mounted with one pair facing forward and one facing downwards. These pairs operate together to allow the vehicle to drive toward the signal using the forward mounted pair and to detect the pinger below using the downward facing pair. While this non-traditional configuration gives an ambiguous solution with uncertainty of the location, an assumption can be made that the pinger will be in front of, and below the vehicle.

3.7 DAQ and Preamplifier

To amplify the signal from the hydrophones to a usable voltage an Analog Devices ADA4610-4 op-amp is used. This amplifier was selected because it gives 4 channels of amplification within one small format SMD chip, and has exceptional noise characteristics. This

chip is well suited to the task because it offers low noise output for high impedance sensor amplification at high frequencies. An Analog Devices AD8054AR-EBZ evaluation board was used. This allowed for rapid development and easy prototyping. This board allows for configuration of analog signal processing. 19dB of gain was selected to achieve a signal at an appropriate voltage to be read by the DAQ. A NI cDAQ-9181 was selected to be the analog to digital converter. This was selected because it can perform 500 kilo samples per second, simultaneously with 16-bit resolution. This DAQ is connected via Ethernet to allow for high speed communications.

4. Software and Controls

The main software packages used on the subs are as follows: Python 2.7, NumPy, OpenCV, and matplotlib, as well as the Arduino IDE. Python was chosen as the primary development tool because it's easy, quick to develop on, and intuitive to read. Many of the members the team acquires only have programming experience in MATLAB, so it was important for us to have a language that is easy to understand. All of the software on the main computer of the sub is written in Python. NumPy is used for array and matrix manipulation. OpenCV is used for vision processing and part of our GUI. Matplotlib is used to graph all of the data the sub records. All of the software is developed on Windows 7, but should be compatible on most platforms, due to the high portability of Python.

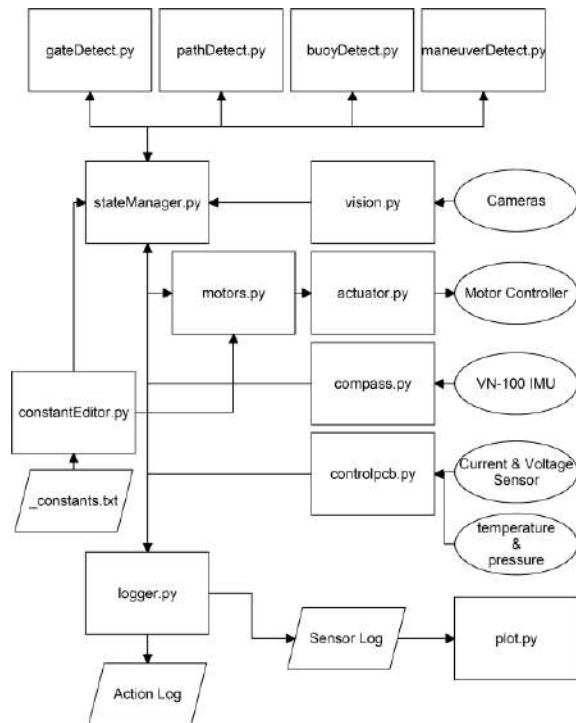


Figure 12 - Software Diagram

4.1 Software Design

YellowFin's software system can be broken down into four categories: hardware communication, data processing, data logging, and value manipulation.

4.1.1 Hardware Communication

The motor controller board is interfaced with through a UDP connection. Through actuator.py, each of the six motors can be independently controlled through i2c. To accomplish this, a command is sent to the Galileo containing the percentage of maximum speed of each of the thrusters. The motors.py script handles higher level movement commands, translating them into individual motor movements.

The motor voltage and current along with the barometric pressure and temperature sensors are read using controlpcb.py, interfacing with the Galileo. This script communicates over UDP to return the sensor value.

Compass.py interfaces directly with the VN-100 IMU via serial RS232. Images are retrieved from the camera using OpenCV.

The daq is communicated with through a LabView program compiled into a dll. This dll connects to the daq through ethernet protocols. Through hydrophones.py, this dll is loaded and the angles computed by each hydrophone pair are returned.

4.1.2 Data Processing

All data from the sensors is processed through the state manager. Through a set of conditional statements and the vision processing scripts (gateDetect, pathDetect, buoyDetect, maneuverDetect) the state manager determines what actions should be sent to motors.py. The vision processing scripts interpret a target heading and depth, and the state manager communicates this to the motor control script which uses the current heading to determine motor power changes.

4.1.3 Data Logging

New to this year's vehicle are action logs and data logs. Writing to any of the logs is handled through logger.py. On startup, both logs are automatically created. The data is written directly to text files, instead of through a buffer, to ensure that if a crash is to occur, the logs are intact.

All action logs names have the format of month, day, year, hour, minute, and second (6_25_2014_16,7,19.txt). Every action ranging from connecting to a communication port to changing the power of a motor is logged with a source and a timestamp. The files are easily viewed and organized in a spreadsheet reader.

The sensor logs function much like the action logs. They are comma delimited text files that are named with the current time in seconds.

On average, the data from every sensor is logged every half second. These logs have been crucial for finding relationships between some of our data such as internal case temperature and pressure.

From several days of testing logs a strong set of data which showed the relationship between internal case pressure and temperature while at the surface. The linear relationship found between these two values allows us to reliably determine our depth even though our barometer reads case pressure.

4.1.4 Value Manipulation

Many of the tasks which are performed require careful tuning which may not be effective in all environments. To facilitate expedient value tuning, the constants used in the motors for calibration, heading and depth control PI controllers, thresholds in vision and more can be altered by loading a text file using `constantEditor.py`.

4.2 Software Algorithm

This section contains the software processes that are used to control the vehicle through the competitions tasks. At a high level a simple finite state machine can model the process. All code described in this section was written in Python, and is composed of in-house written scripts and libraries.

1. Load waypoints folder containing each point of interest, and the angle between them. This information is pre-entered based off the condition of the pool.
2. Using file as a reference, head in the direction of the next waypoint, while using the cameras to attempt to lock onto the object. If the target can be seen, head towards it. If not hold heading and continue forwards. A simple PI controller is used to maintain heading. Additionally,

depth is also monitored though each control cycle. If the measured depth is out of the accepted range, a correction is made using a P- controller.

3. If waypoint is located, run the script associated with that task, update the waypoint location, and return to the main control loop.

4.3 Vision Processing

It was determined that vision processing is the best solution to object detection. The onboard cameras were selected due to their accurate color detection. This decision was based off of many years of research using varying types of cameras. By having an image with few color defects, a threshold can easily be applied. To increase the color accuracy of the image, the OpenCV is used to normalize the histogram. This ensures that the lighting conditions between images are consistent. A binary threshold is then applied to the Hue, Saturation, and Value (HSV) planes of the image. Once a binary threshold is applied, contours are used to get an array of onscreen objects. Once a list of objects is made, they are sorted by area, location, and shape. This method of computer vision can be seen in Figure 14. In addition to testing vision algorithms on new images, we use past footage from the Transdec pool. This is done due to the water clarity differences between the competition pool and the pool at Embry-Riddle. The top image demonstrates our algorithms ability to find the gate. The bottom image shows the usage of contours. The outline of found images is shown.

The vision software was tested and modified in several varying lighting conditions. This allows for us to be able to tune the environmental variables so that the algorithms to work in nearly all conditions.

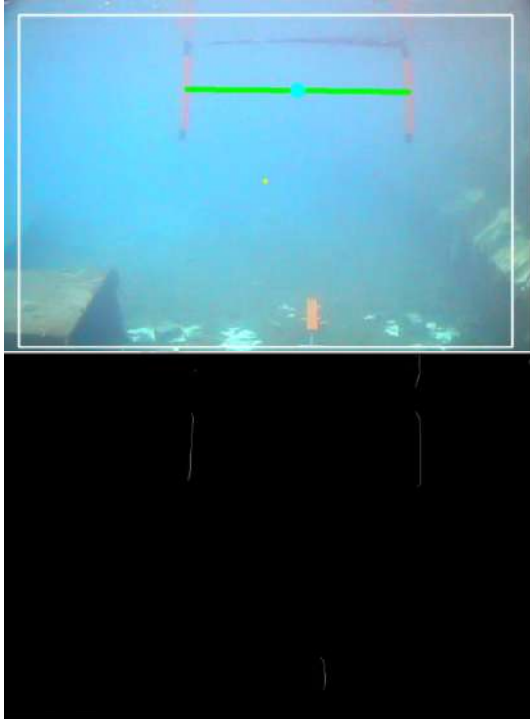


Figure 13 - Detected Object (Top) Threshold Contours (Bottom)

5. Remote Debugging

For testing purposes YellowFin can be remotely controlled. An Ethernet tether can be connected to the vehicle to use Window's Remote Desktop application to log into the sub's main computer. Once connected the vehicle can be controlled, as well as viewing the output from cameras and sensors through our self-made GUI.

6. Conclusion

YellowFin is a fully autonomous underwater vehicle designed and manufactured by engineering students at Embry-Riddle Aeronautical University. In addition to being designed to compete in the AUVSI RoboSub competition, YellowFin was also designed to be a research platform. We believe that YellowFin is an excellent demonstration of systems engineering. Through careful planning and implementation all hardware and software systems were able to interact together with no problems. Through the design and development of this platform, we as a team were able to grow significantly.

7. Acknowledgments

We would like to thank Embry-Riddle Aeronautical University's College of Engineering specially the Department of Electrical, Computer, Software, and Systems Engineering and the Mechanical Engineering Department for their continued and substantial support of Team YellowFin. We would especially like to thank our faculty advisors for all their support, encouragement, and advice. We would also like to thank all of our sponsors for the outstanding support and donations that have made this project possible.

8. References

VectorNav. (2014, June 26). *VN-100 Rugged IMU / AHRS*. Retrieved from VectorNav: <http://www.vectornav.com/products/vn100-rugged>